Dynamic Modeling of Routing Protocol Using Colored Petri Nets

Garimidi Hareesh

Roll. 213CS3183

under the supervision of

Prof. Suchismita Chinara

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769008, India

Dyanamic Modeling of Routing Protocol Using Colored Petri Nets

Dissertation submitted in

May 15

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

Garimidi Hareesh

(Roll. 213CS3183)

under the supervision of

Prof. Suchismita Chinara

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

May, 2015

Certificate

This is to certify that the work in the thesis entitled *Dynamic Modeling of Routing Protocol Using Colored Petri Nets* by *Garimidi Hareesh*, having roll number 213CS3183, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering Department*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. Suchismita Chinara
Assistant Professor
Department of CSE
NIT, Rourkela

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Suchismita Chinara, who has been the guiding force behind this work. I want to thank her for introducing me to the field of Colored Petri Net and giving me the opportunity to work under her. Her undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without her invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to her for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I would also like to thank all faculty members, PhD scholars, my seniors and juniors and all colleagues to provide me their regular suggestions and encouragements during the whole work.

At last but not the least I am in debt to my family to support me regularly during my hard times.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Garimidi Hareesh

Abstract

The growth of interest and research on mobile ad-hoc networks is exponentially in recent years. In a Mobile Ad hoc NETwork (MANET), Nodes are mobile in nature, so the node movement in the dynamic environment causes frequent topology changes to the network. In this paper, we are going to model the AODV (Ad hoc On-demand Distance Vector) routing protocol and analyse the STATE SPACE diagram of AODV routing protocol using CPN TOOL to detect the problems in routing protocol and resolve the issues before implementation. Modelling in CPN tools require predefined input values to be incorporated in the states which are used to detect the neighbours and track the path from one node to another node on the network. In this model, we assume all nodes have sufficient energy. State space diagram helps to identify the loops, path breaks and dead nodes in the network. In this paper we done dynamic modelling of AODV routing protocol using cpn with the help of NS2 and MATLab. In this process we chosen a trace file from ns2 convert it into cpn input format with the help of MATLab.

***Keywords*:** AODV,State Space, Mobile Ad-hoc Network, Coloured petrinet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mobile ad hoc network (MANET) is a network which is infrastructure less. This network consists of mobile nodes and wireless machine nodes connected to each other [1]. There is no centralized administration mechanism in MANETs. In MANET every node acts as a router. As the name indicates, Mobile means the nodes are not stationary, Ad hoc means no fixed infrastructure. Therefore a network with no fixed topology or infrastructure is called adhoc network. In MANET, a node can send packets directly to another node if both the nodes are in their respective transmission ranges or else packet transmission can be mutlihop [1]. Due to arbitrary movement of nodes, network topology changes rapidly. Since the nodes are changing much rapidly it is very dificult for the routing protocols to perform their task correctly. Hence a topology approximation mechanism, known as Ad hoc On-demand Distance Vector (AODV) routing protocol, is used to perform simulation of a typical routing protocol which addresses the problem of mobility [2]. AODV routing protocol [3] is a reactive routing protocol, which means that route from source node is created in an on demand basis.

1.1 MANET Characteristics:

Software testing is the process which identifies the bugs, errors and faults of a system to ensure a better software product. Its objective is to provide independent information about the product to the user and also the risk of failure of the product. Software testing responsible is to provide better quality software. Its aim is to test maximum no. of test cases out of infinite to cover all the feasible solutions with intend to find maximum bug inside the program. It takes consideration of both normal as well as abnormal condition for the identification of unexpected errors at any condition. It is an pessimistic approach for generation of test cases .

- In MANET, every node can act as both host as well as a router. So, it has an autonomous behavior.
- Whenever the source and destination are not reachable, MANETs are capable of multi-hop routing.
- It has a circulated nature of operation for security, directing and host setup. An unfired firewall is missing here.
- The hubs can join or leave the system at whatever time, making the system topology changing in nature.
- Versatile hubs are described with characteristics such as light weight, less memory and power.
- The unwavering quality, productivity, security and limit of remote connections are frequently subpar when contrasted and wired connections. This shows the uctuating connection transfer speed of remote connections.
- Mobile and spontaneous conduct which requests least human intercession to arrange the system.
- All hubs have indistinguishable characteristics with comparative obligations and proficiencies and thus it structures a totally nature.

- High client thickness and extensive level of client portability.
- Nodal integration is discontinued.

Here are some of the applications of mobile ad hoc networks:

- **Military Services:** Military Services are one of the most discussed and common application area of mobile ad hoc networks where installation of any fixed infrastructure is not possible in the enemy territories or inhospitable terrains. In this environment MANET provide the required communication mechanism in no time. Here, the soldiers are considered to be the mobile nodes. So the network is required to remain connected even though the soldiers move freely. This support is provided by the MANET. Another application in this area can be the coordination of the military objects and the personnel in the battle field.
- **Emergency Services:** In certain situations that are unexpected and unavoidable like search and rescue, crowd control, disaster recovery and commando operations, the use of mobile ad hoc networks is very much suitable. The major factors that favour the deployment of MANET in such situations are its self-configuration with minimum overhead, unavailability of fixed or centralised infrastructure as well as freedom and flexibility of mobility of the nodes. Since the ad hoc networks require minimum initial network configuration, it can also be deployed in situations where conventional infrastructure based communication is disturbed due to natural calamity or any other reason.
- **Collaborative and Distributive Computing:** The use of mobile ad hoc network is very much necessary in such situation where a group of researchers want to share their research findings or share their research materials during a conference or on?the?fly.
- **Sensing and Gaming:** Sensor network is a special case of ad hoc networks where mobility is generally not considered. However the battery power is a key

factor in sensors. Each sensor is equipped with a transceiver, a small microcontroller and an energy source. The sensors relay information from other devices to transport data to a central monitor. The sensors are used to sense the environmental condition such as temperature, pressure, humidity etc.

- **Personal Area Networking:** Personal communicating devices like laptops, PDAs, mobile phones create a network to share data among each other called the personal area network (PAN). The PAN cover a very short range for communication and can be used for ad hoc communication among the devices or for connecting to a backbone network.

1.2 Petri nets :

Petri nets were presented by Dr. Carl Adam Petri in 1962 [4]. These are a capable demonstrating system in software engineering, framework designing and numerous different controls.Petri nets combine a graphical representation of the dynamic behavior of systems with well-defined mathematical theory. Modeling and analysis of the system is done by the theoretical aspect of Petri nets, while the visualization of the modeled system is done by the graphical representation. This combination of theory and graphics is the main reason for the great success of Petri nets [5]. husly, Petri nets have been utilized to model different sorts of element occasion driven frameworks like PCs systems, correspondence framework, constant registering framework and so on.

A Petri net (otherwise called a spot/move net or P/T net) is one of scientific displaying dialects for the depiction of conveyed frameworks. A Petri net comprises of spots, moves and curves. Circular segments run from a spot to a move or the other way around, never between spots or between moves. The spots from which a bend races to a move are known as the information spots of the move; the spots to which curves run from a move are known as the yield spots of the transition. Graphically, puts in a Petri net may contain a discrete number of imprints called tokens. Any conveyance of tokens over the spots will speak to a design of the net called a stamping. In a unique sense identifying with a Petri net graph, a move of a Petri net may fire on the off chance that it is empowered, i.e. there are adequate tokens in every last bit of its data places; when the move fires, it expends the obliged info tokens, and makes tokens in its yield places. A terminating is nuclear, i.e., a solitary non-interruptible step.

A place in a petri net contains a single and multiple tokens as shown in Figure-1.1.

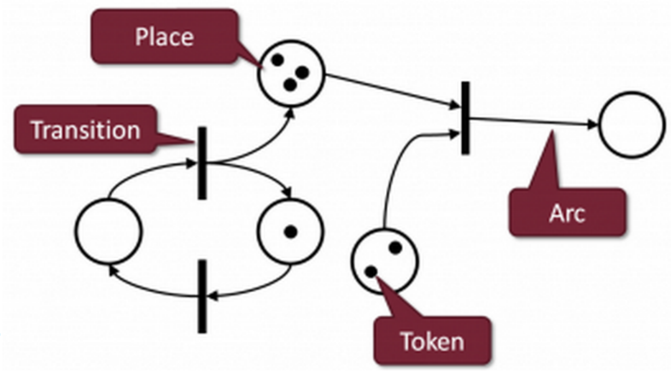


Figure 1.1: simple petri net diagram

1.3 Colored Petri nets

Colored Petri Net [6] provides a framework for construction and analysis of concurrent and distributed systems [7]. A Coloured Petri Net model describes the states(the places) that a system may obtain and the possible transitions in between them. The strength of CPNs over traditional Petri Nets is that, it supports hierarchy, colour, and time in the model. Hierarchy in the CPNs indicates that the models can be structured in a number of related modules. This concept is based on the concept of hierarchical structuring of the programming language [8], that supports the bottom-up or top-down style.

Modules made can be reused in a few sections of the CPN model and further sub-modules can be made from it [9]. The modules of the CPNs are called pages. A mind boggling model can have the same number of as many pages like a protracted and complex program, that is isolated into a few modules. In progressive CPNs, a move and its related parts make a connection to another CPNs giving a more exact and itemized portrayal of the movement spoke to by the move. Such moves are known as the substitution moves. The chain of importance engraving in the substitution move characterizes the points of interest of the substitution in divided modules called the subpages [10]. The spots in a sub-page are stamped with an info tag In tag,output tag Out label or data/yield label I/Otag. These spots are known as the port spots [10]. They constitute the interface through which the sub page speaks with its surroundings.

The sub page receives tokens from its surroundings through the input port. It delivers tokens to its surroundings through the output ports and the I/O port communicates to its surroundings in both ways. The places associated with a substitution transition are called the socket places. The port spots of the sub pages are identified with the attachment spots of the substitution move by giving the port assignments [practitioner's guide]. At the point when a port spot is alloted to an attachment put, the two spots get to be indistinguishable. The port spot and the attachment spot are two distinct representations of a solitary applied spot, i.e. the port and the attachment spots have constantly indistinguishable markings. At the point when an info attachment gets a token from the surroundings of the substitution move, that token additionally gets to be accessible at the information port of the sub-page [10], and subsequently the token can be utilized by the moves on the sub-page. Likewise, the sub-page may deliver tokens on a yield port. Such tokens are likewise accessible at the comparing yield attachment and thus they can be utilized by the surroundings of the substitution move.

Communication is important for modeling of AODV routing protocols. Systems where concurrency, communication and synchronization play an important role can be modeled using Colored Petri Net tool [2]. As has been done in (Gordon 2001), one of the approach to ensure the correctness and validity of existing routing protocol is creating a formal model for the protocol [11] and analysis of model to determine if the existing protocol provides the defined services correctly or not. By verifying the routing protocol using formal modeling, one can gain CPN is a graphically oriented language [12] for design, specification, simulation and verification of systems. An advantage to use CPNets is that it is possible to use the same model for performance analysis as well as for checking the logical and functional correctness of a system [2].

Another idea of progressive CPNs [13] is the combination places. This shows that various individual drawn spots can be considered i.e. they all speak to a solitary reasonable spot. At the point when a token is included or uprooted at one of the spots, an indistinguishable token will be included or evacuated at all different places in the combination set. So it is pass that the relationship between the individuals from a combination set is like the relationship between two spots which are alloted to one another by a port task. At the point when all individuals from a combination set have a place with a solitary page and that page has stand out page case, place combination is just an attracting accommodation to evade an excess of intersection bends in the model. Anyhow, the circumstance is complex and intriguing when the individuals from a combination set have a place with a few diverse subpages or to a page that has a few page examples. The different sorts of combination sets are the worldwide combination sets, page combination sets and occasion combination sets. The worldwide combination set can have individuals from distinctive pages though the page combination set and case combination sets just have individuals from a solitary page. Hues connected with every spot in the CPNs focus the sort of information it may handle [14].

The types of the places shown in Figure-1.2 are similar to the types in programming languages. It can be a complex type as the record which may contain heterogeneous data types. The color set is usually defined as: colset No = int; Where COLSET is a keyword to declare the color set, NO is the name of the colour set and int indicates that this colset can have integral values as tokens. The state of a CPNs is called as its state that shows the number of tokens distributed on the individual

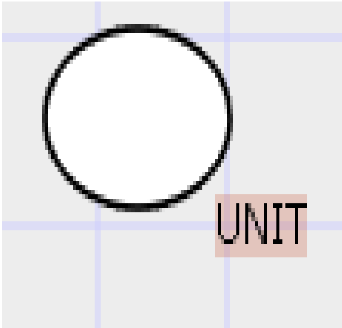


Figure 1.2: Place inscription: place type

places. Each token carries a value that belongs to the type of the place on which the token resides. The tokens present on a particular place denotes the marking of that place. The initial state of a place is denoted as its initial marking. It is usually written in the upper left or right of the place as shown in Figure-1.3.

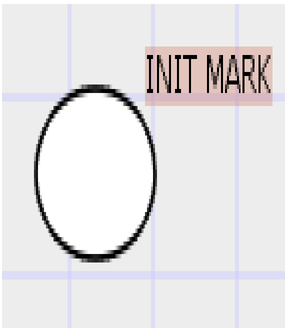


Figure 1.3: Place inscription: initial marking

Every place in coloured petrinet is connected to a transition which fires the tokens from one place to another. No two places and transitions can be connected.The place and transition are connected through arcs and an arc has an inscription which may contain a condition or an expression as shown in Figure-1.4.

The inscription of transition are shown in Figure-1.5. Figure-1.5(a) shows how time values are added to the transition. Guard functions are written in top left corner of the transition as shown in Figure-1.5(c). Figure-1.5(b) shows the code segment where the inputs and outputs are written and the code written action part

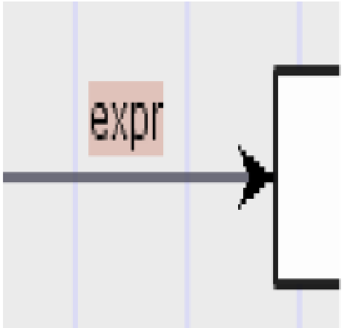


Figure 1.4: Arc inscription

is performed whenever a transition is enabled.

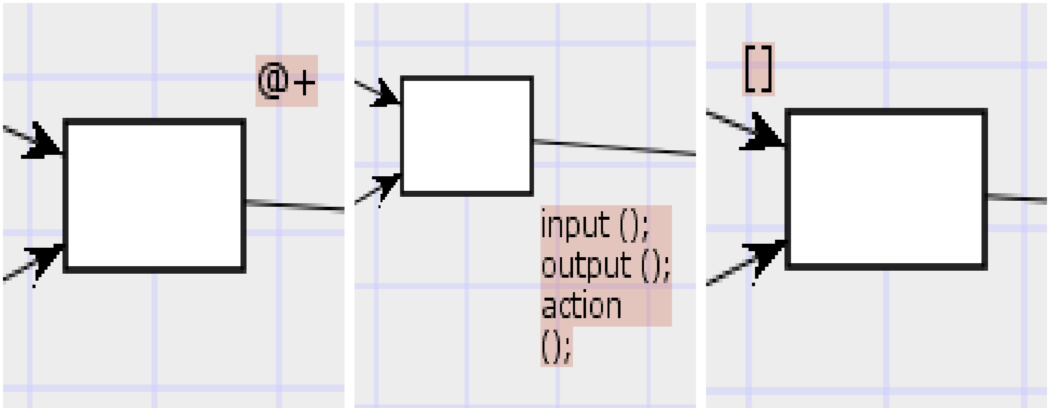


Figure 1.5: Transition inscriptions:time,code segment,guard

A transition is enabled at that time only when all the inputs to a transition are available. A green aura is created around the rectangular box whenever a transition is enabled as Figure-1.6.

As a scientific apparatus, a petri net model [15] can be depicted by an arrangement of direct mathematical statements or other numerical models mirroring the conduct of the framework. This opens a plausibility for formal investigation of the model. A formal check of properties could be possible identified with the conduct of hidden frameworks e.g. priority relations amongst occasions, simultaneous operations, suitable synchronization, free from gridlock, dull exercises and shared avoidance of imparted assets to some degree. Timed petrinets gives a uniform domain to

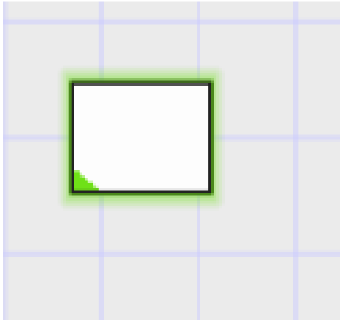


Figure 1.6: Enabling of a transition

displaying, outline and execution investigation of discrete occasion frameworks.

1.4 Timed petri nets

A timed CPN [16] can display the amount of time certain exercises require and the amount of time goes between different exercises. In many cases,its lacking to model the normal measure of time a certain action takes- it is important to incorporate a more exact representation of timing of the framework. Bringing time origination into the CPNs can be reclassified as timed CPNs. This presents the idea of worldwide clock.The clock value which is either discrete or continuous represent the model time. In the timed CPNs, each token carries a time value called the time stamp. The time stamp depicts the soonest model time at which the token can be utilized, that is it can be uprooted by the event of a coupling component. In a timed CPNs a coupling component is said to be shading empowered when it fulfills the empowering standard for un-timed CPNs. On the other hand, to be empowered, the time stamps of the tokens to be uprooted must be not exactly or equivalent to the present model time. Figure-1.7 shows how timing qualities are consolidated in a move.

The stamping of a spot where the tokens convey a period stamp turns into a timed multi-set determining the components in the multi set together with their number of appearances and their time stamps. The timed shading sets are proclaimed as: colset INTDATA =product INT*STRING timed; and the possible marking of a place

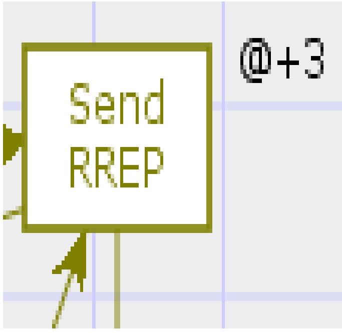


Figure 1.7: Adding timing values to a transition

with timed token is as: 2‘(1,colour)@[19,45] This indicates the marking contains two tokens with value (1, colour) and time stamps 19 and 45 respectively. The @ symbol can be read as at and the symbol [ ] is used to specify the time stamps.

1.5 state space analysis using cpn tool

CPN incorporates StateSpace(SS) devices. This gives the focal point to move between the test system and the SS device. SS can be reviewed in the test system. This serves to picture the stamping straightforwardly on the graphical representation of the CPN model. You can see the empowered move cases, research their ties and make recreations. Similarly, when a checking has been found in the test system, it can be added to the state space or utilized as the introductory stamping for another state space.

The SS device has an extensive number of implicit standard questions. They can be utilized to explore all the standard properties of a CP-net, for example, reachability, boundedness, home properties, liveness and reasonableness. Notwithstanding the standard questions there are various intense inquiry offices permitting you to figure your own, non-standard inquiries. The standard inquiries oblige no programming by any means. The non-standard inquiries normally oblige that you compose 2-5 lines of truly direct ML code.

1.6 Thesis Layout

The rest of the thesis is organized as follows:

1. Chapter 2: Related Work In this chapter, literarture survey is done in AODV, Coloured Petri Net tool, Timed petri net and state space analysis.

2. Chapter 3: Colored petri net modeling of AODv protocol for verification and to build a state space diagram. This chapter presents a model of AODV routing protocol implemented in Coloured petri net tool. The protocol is simulated and hence validated using timed petri net concept. Here end to end delay is calculated. State space diagram implementation and report generation.

3. Chapter 4: Simulation in Timed Petri Net with help of NS2 In this work, the existing AODV routing protocol is simulated in NS2 tool and using generated trace file, necessary information for calculating delay value is extracted. These values are feeded into the CPN tool after the calculation of end to end delay..

4. Chapter 5: At last we concluded in this chapter.

Chapter 2

Literature Surve

Ad-hoc On-Demand Distance Vector (AODV) Routing is a directing convention for versatile specially appointed systems (MANETs) and different remote impromptu systems. It is mutually grown in Nokia Research Center, University of California, Santa Barbara and University of Cincinnati by C. Perkins, E. Belding-Royer and S. Das. The hubs in AODV continue changing their position rapidly hence making it hard to model the framework.

Hued Petri Nets has been turned out to be a capable apparatus for reproducing and examining the non-determinism, concurrency and diverse level of reflection of any correspondence convention. Hued Petri Nets have been utilized by a portion of the specialists for approving and displaying a percentage of the highlights of the portable impromptu systems. Chinara et al. [6] have proposed the approval of neighbor identification convention for specially appointed system by utilizing the CPN devices.

Erbas et al. [7] proposed a two outlined position construct directing methodology based model in light of Colored Petri Nets for versatile specially appointed system. Here the creator demonstrates that the multicast steering convention conveys preferable result over the fundamental ODMRP (On Demand Multicast Routing Protocol). This model (CPN model) created dependable unicast and multicast directing strategy in light of land position of a hub.

Kodikara et al. [9] proposed the recreation model of setting trade taking into account various leveled hued petri nets. Reenactment of vertical correspondence model has done which is cross layer data trade module of connection trade (conEX).State space utilized for check of Petri nets dynamic behavioral properties like liveness, home property, boundedness and decency.

To enhance nature of course determination for MANET directing convention, Nakhaee et al. [17] exhibited new course determination criteria rather than bounce check. This plan adds two parameter to course ask for bundle: number qmean and number maxq, demonstrates the normal line length and greatest line length of the hubs in the way individually. Nakhaee et al. [17], exhibited a way to deal with enhance AODV convention directing unwavering quality. The methodologies of both the papers ([17]) are practically same. mohamed et al. Versatile impromptu system has a few charecteristics which secernate it from traditional one.Nodes in the system suppots the system infrastruture and security obliged when the parcels are passing.Author proposed security methodology in view of immunty and multiagent paradigm.It portray the how the portable specially appointed system is secured by distinctive operators coordination.It has been examined that much work has not been done on the portability example of the impromptu systems by utilizing CPN devices. This gives an inspiration to the present work where CPN devices have been utilized for the displaying of the Random Way Point portability for MANET [18]. The paper [12] addresses the mobility peoblem.Colored petri nets used as a simulation tool , which simulate the network without knowing the network topology.Colored petri net is a tool which provide great perceptivity to modelling a mobile ad-hoc network protocol.The author finds the route in the simulation part.This paper compares the DSR and AOMDV.For small network AOMDV has lesser delay to find out routes and in the other hand DSR has better efficiency than AOMDV but comparitively high delay in finding the routes.

Further to improve the TCP performance over MANET, Xiong et. al. [Xiong, Yim, Leigh and Murata] have proposed a reactive approach TCP-MEDX to detect the causes of packet loss. As mobility of nodes is the biggest challenge in MANET, the round trip time is replaced with an average propagation delay for indicating congestion. The authors claim that the TCP-MEDX mechanism is able to detect the packet loss much accurately. Xiong et. al. have created a formal CPN model of the very well known routing protocol for MANET, the Ad hoc On-Demand Distance Vector (AODV) to analyse its correctness in service [17]. To meet the challenge of dynamism of the nodes, the authors have proposed a topology approximation (TA) mechanism. The TA mechanism works with certain assumptions. They are:

1. All the nodes in the MANET have equal transmission range.

2. Every node in the MANET has the same number of neighbors which is equal to the average degree of the MANET graph.

Here, the second assumption is not a realistic approach. Because in MANET, the node movement frequently changes the degree of connectivity among the nodes and also the network topology. In such a non-deterministic environment,the mechanism of topology approximation may not be possible. Further, the dynamic operation of MANET using CPN is illustrated by Yuan et. al. in [17]. The CPN tools are used by the authors for the elegant and simple modelling of the protocol without using any assumption or approximation. By using the formal specification and verification method of the modeling tool, the authors could be able to find the errors existing in the protocol and suggested the modification to eliminate those errors. There is very few work done in simulation of AODV protocol using timed Petri nets [9]. An extension to AODV (known as MAODV) was made in [3] [4] to support multicast routing in MANETs. In conventional AODV routing protocol, every destination node maintains a unique sequence number. In multicast routing algorithm periodic broadcast of HELLO messages in a network is done by a group leader amongst the nodes (which is by default the first node in the network until it moves out of network otherwise any random node can be chosen). The proposed algorithm creates a multi cast tree [19] in an on demand basis and is constructed in the same way as that of route discovery process. Due to this tree formation loop is prevented.

Chapter 3

Dynamic Modeling Of AODV Protocol Using State Space in CPN

This chapter deals with modeling of AODV routing protocol using Coloured petri net.

3.1 Ad hoc On demand Distance Vector Protocol

AODV is a routing protocol [20] defined for MANETs and other wireless ad-hoc networks [21]. Routes for new destinations are quickly discovered by the mobile nodes in MANET and they also respond to link breakages and changes in network topology [5]. Whenever there is a link breakage, effected nodes are notified so that they invalidate the routes using that link. In order to identify the most recent path, AODV makes use of a destination sequence number [21]. In order to build the routes in AODV routing protocol, Route request and Route reply query cycles [17] are used. As has been done in (Gordon 2001), creating a formal model for the routing protocol and analyzing the model to determine if it provides the defined services correctly is one of the approach to ensure the correctness of an existing routing protocol.

3.2 AODV Terminology

- **Active route/valid route:** A route is said to be active route if it has a corresponding entry in the routing table entry marked as valid. For transmitting data packets only active routes in the network can be used.
- **Destination:** It is an IP address to which the data packets should be transmitted. A node identifies itself as the destination node by looking upon the appropriate field in the packet.
- **Forwarding node:** It forwards the packets which are destined for another node. This node retransmits packets to a next hop along a path that has been set up using routing control messages.
- **Forward route:** A route set up which is used to send data packets from a node towards its desired destination by following the route discovery process.
- **Invalid route:** It is denoted as invalid in the routing table entry. This route is expired. It cannot be used to forward data packets, but it helps in providing information useful for route repairs, and also for future RREQ messages.
- **Originating node:** This node is responsible for starting an AODV route discovery process and possibly retransmit the message by other nodes in the network.
- **Reverse route:** This route is used to send the RREP message to the source once the destination is found. Any intermediate node which has the destination's address can invoke this.
- **Sequence number:** It is an automatically increasing number maintained by each node. It is used to determine the freshness of the information that is originating from the originating node.

3.3 AODV Process

As AODV is reactive protocol i.e. the route is created only when it is required. Packet transmission in AODV routing protocol can be multi hop i.e. using other nodes as relay points, a node can send packet to another node beyond its transmission range and thus a node can function as a router. Every node in the network maintains its own neighbor table.

Before sending a data packet every node N first checks its routing table to see if there is a valid route to the destination or not. If it has, it sends the data packet along this route immediately. Otherwise, the route discovery procedure starts which is as follows:

- **Route discovery:** RREQ (Route Request) packet is broadcasted throughout the network if the route between source and destination is not available. As soon as a node receives a RREQ packet, it first checks whether it has received this packet earlier. If yes, then the node simply discards the packet and if not, a reverse routing entry towards the originator of RREQ packet is created. This route can be used to forward route reply later on. If any intermediate node has a valid route towards the destination node, it unicasts a RREP (Route REPly) packet towards the source node. A node on receiving RREP packet creates a reverse route entry towards the originator of RREP packet [13].

- **Route maintenance:** In order to indicate its presence, every node in the network periodically broadcasts HELLO messages to its neighbors. A particular route is marked as invalid if a node does not receive a HELLO message from its neighbor and the node is considered to be exhausted or moved away from the network. Hence a RRER packet is sent to all the nodes and the routing table is updated [13]. In order to get rid of loops, a sequence number is maintained by every node in the network. This sequence number is incremented by the node every time a packet is sent and is stored along with the route information in the route table. It is sent along with RREQ (for source) and RREP (for destination). The most recent path to the destination is indicated by the node with larger sequence number. Therefore such nodes are always preferred [13].

.

If a source node wants to send a data packet, it first checks the routing table for a valid path to the destination. If it is available, then the packet is forwarded through that route otherwise RREQ is broadcasted. This packet contains source ID, sequence number of source broadcast ID, sequence number of destination, previous ID, hop count and destination ID. The purpose of destination sequence number is to prevent the loop in the route discovery process. Every node has its own sequence number which is incremented by one every time there is a link breakage [2]. A node upon receiving this packet checks whether it has received this packet earlier, if it has it simply discards the packet otherwise a RREP packet is unicasted to the source along the reverse path to that of RREQ packet [22]. Setting of a forward path entry to the destination is done when an intermediate node receives this RREP packet. AODV protocol has following 4 states:

- **Routcheck:** It checks the routing table [23] to find whether the source node has an unexpired path to the destination node or not. This is done using the guard has validRoute().

- **RREQInit:** Its main function is to initiate the RREQ message when necessary. The RREQ message is rebroadcasted using the function arc rebroadcast () if necessary.

- **RREQProcess:** This subpage initiates the RREP process if there is a route to the destination, or if there is no route it forwards RREQ message. This functionality is achieved by arc newBID() and arc initiateRREP().

- **RREPProcess:** This subpage plays an important role in the updation of the routing table and also to forward the RREP message if necessary. To achieve these functionalities, two functions arc updateRoute() and arc forwardRREP() are used. RREQInit, RREQProcess and RREPProcess are three substitu tion transitions. If a node desires to send data packet, primitively it enters Routcheck state to check for an existing path. If there is no existing route in the routing table, the node enters RREQInit state and initiates route discovery process i.e broadcasting of RREQ packet [2].

Since there is no clear cut method to design a protocol [24], the designers should utilize those tools which help them in validating the operation of the protocol. The examination of each possible usecases can be done with the help of design verification tools. They are also useful to validate the operation of the protocol in every situation. Since enumerating all possible usecases manually is very difficult, the design verification tools play an important part.

3.4 Dynamic Modeling and Simulation of AODV in CPN

Hierarchical colored petri nets are used for the simulation of AODV routing protocol. Codes are written in arc, place and transition, which are known as arc inscription, place inscription and transition inscription. These all inscription are written in CPN ML programming language. The Figure-3.11 shows the main module of AODV. It contains the set of information which a node carries. It contains: (Source ID, Destination ID, Broadcast id, self id, Store, hop count, Destination Sequence No).

The whole data is transmitted to the place node when the transition generateres. Then this data is transmitted to the substitution transition Protocol.

The Figure-3.2 shows the subpage of the Protocol Module. This subpage contains, four places named node, RREQ, Routing table, OUT and four substitution-transitions namely CheckRoute, RREQInit, RREQProcess and RREPProcess.

The subpage of the CheckRoute substitution transition is shown in Figure-3.3. Here, firstly in order to check if there is a valid route between the source and destination the routing table is checked.

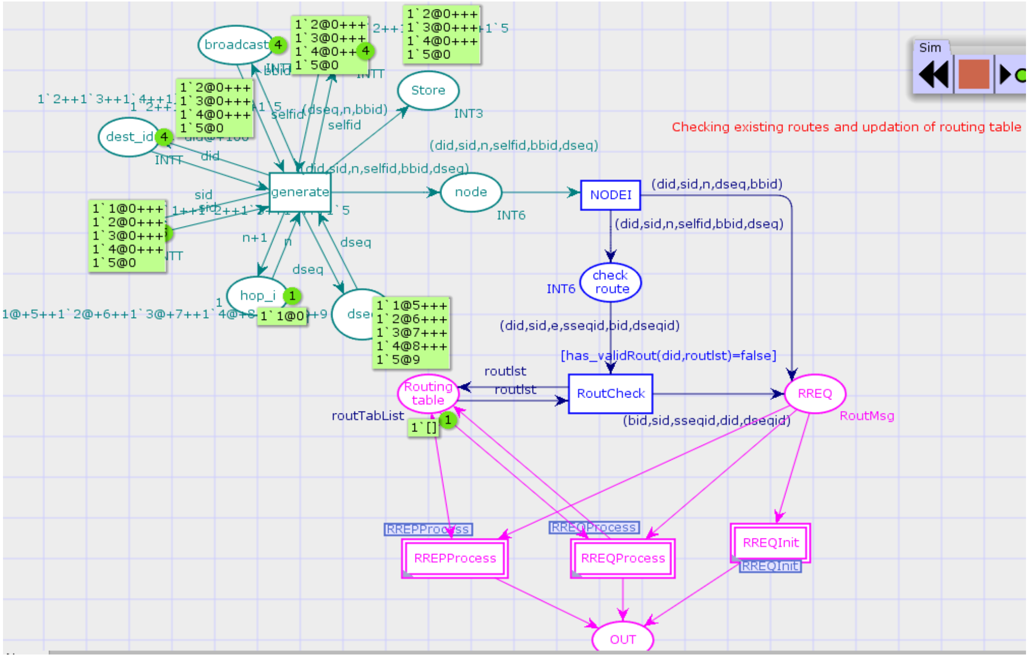


Figure 3.1: The main page of the simulation

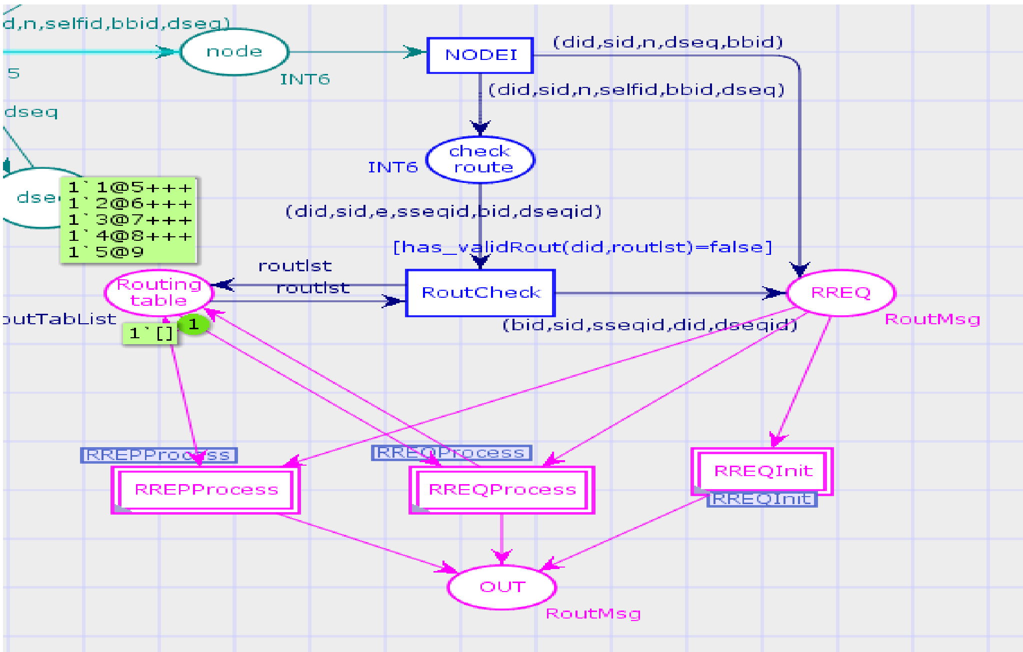


Figure 3.2: Protocol Module

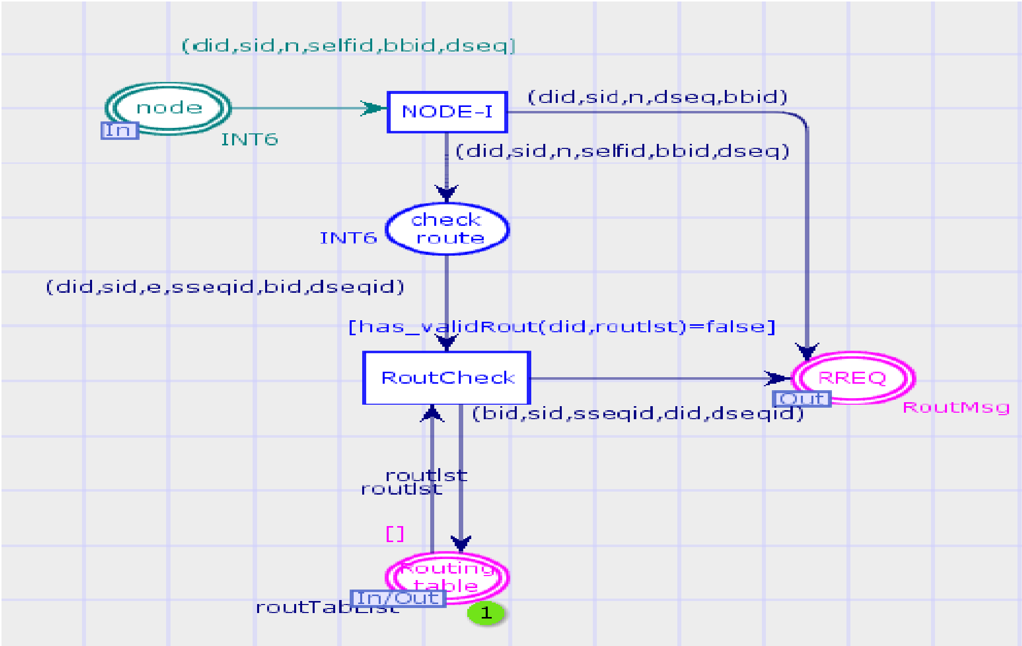


Figure 3.3: Check Route Sub Page

As shown in Figure-3.4, the packet is sent using that route if path exists. If the route doesn't exist then it enters RREQInit state and route discovery process starts. The source node sends RREQ packet to its neighboring nodes.

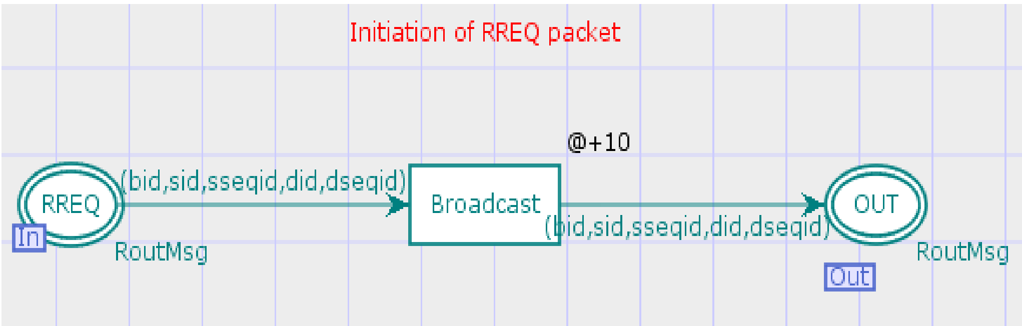


Figure 3.4: RREQInit Subpage

The Figure-3.5 shows the RREQProcess subpage. Whenever a node receives RREQ packet, it checks whether it is the destination node or not (looking upon the destination field in the packet header). Hence two paths are shown in Figure-3.5

This page contains the following transitions:

- **BIDcheck:** this transition checks whether the broadcastid received is fresh one or not. It takes 10 units of time delay.
- **Broadcast:** If the current node is not the destination node then it simply forwards the RREQ packet received to its neighbors. This condition is checked using the guard function guard broadcast(). It takes 2 units of time delay.

send RREP: This transition initiates the RREP packet when the node itself is the destintion node. This functionality is checked using the guard function guard sendRREP() and the RREP packet is initiated using the function arc initiateRREP().

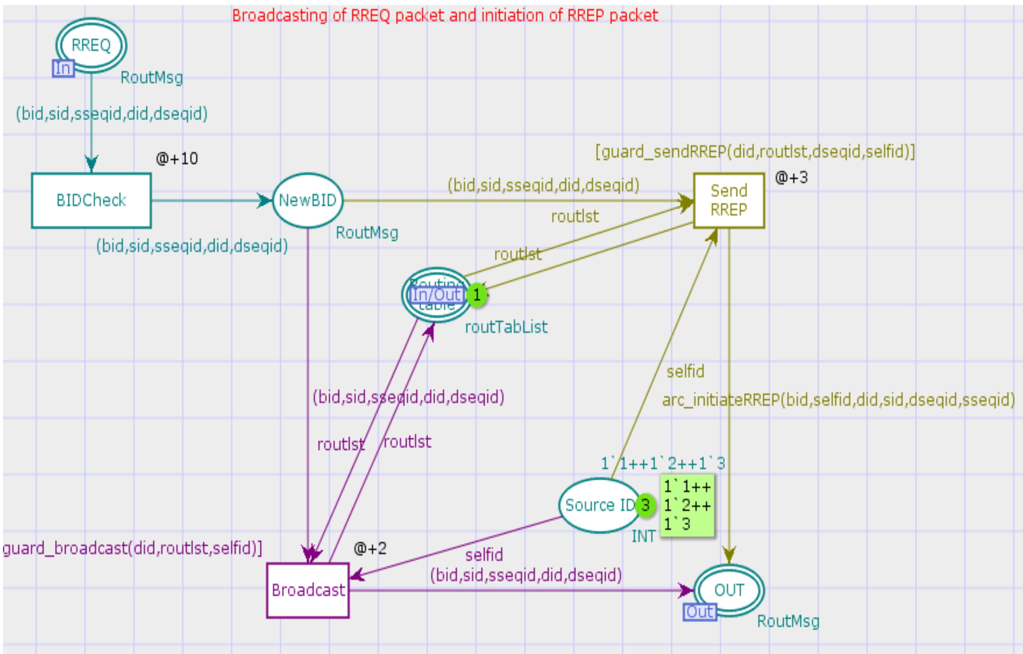


Figure 3.5: RREQProcess Subpage

The RREQProcess subpage has the following places:

- **RREQ:** It transmits the packet received from RREQInit page to this page.
- **NewBID:**Contains all the tokens which haven't received earlier.

- **Routing table:** Provides the routes using the list routlst to transitions Broadcast and Send RREP.
- **Source ID:** Provides the source id to both the transitions(Broadcast and Send RREP)
- **OUT:** Contains all the tokens.

The Figure-3.6 shows the RREPProcess subpage. RREP packet is unicasted along the route reverse to that of RREQ packet. The page shown contains the following main transition :

- **AddRout:** The reply packet contains the route from source node to destination node hence this route is added to the routing table using the function arc updateRout().

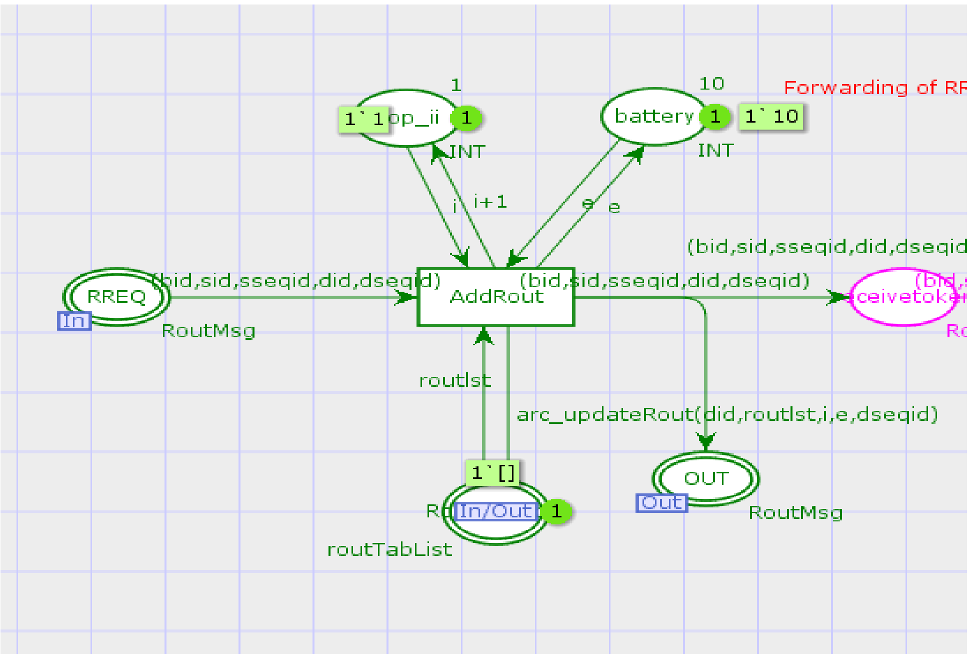


Figure 3.6: RREPProcess Subpage

This page contains the following places:

- **RREQ:** It transmits the packet from RREQProcess subpage to add rout transition in this page. Hopi: It gives the hop count.
- **Battery:** It assigns a very large value as battery.
- **Routing table:** It stores the routes using which data packet is sent. It updates the routing table.
- **OUT:**It contains all the information about the routes.

The simulation process can be explained as follows:

Initially the transition generate contains all the attributes of a packet i.e. source id, destination id, braodcast id, destination sequence id and hop count as shown in Figure-3.7.

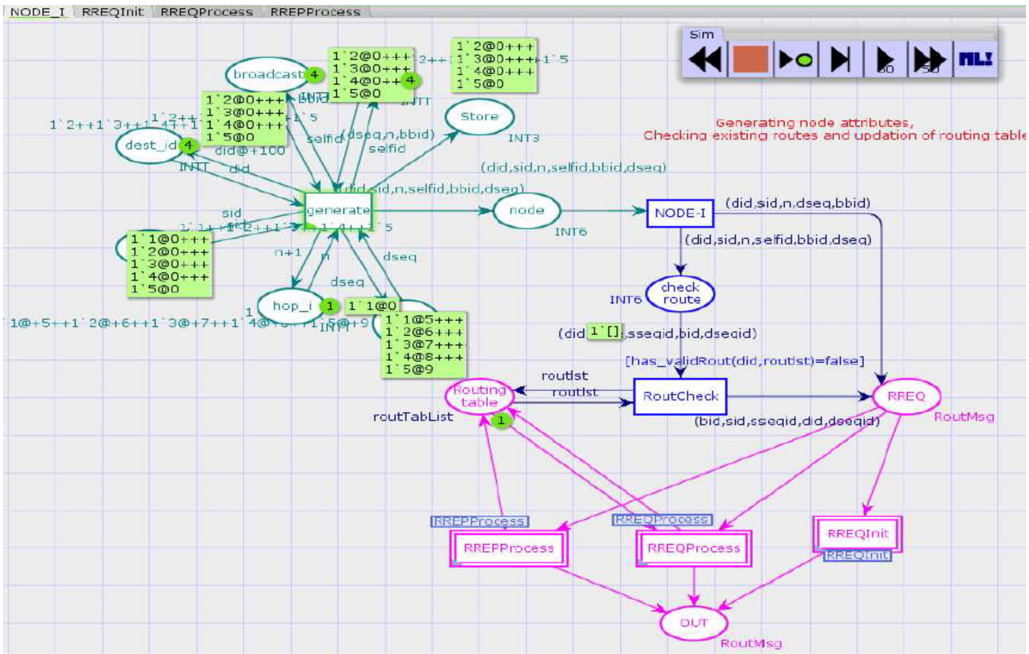


Figure 3.7: start of simulation process

The information generated about a packet is transmitted to the transition NODE?I. Thereafter routing table is checked for an existing entry.

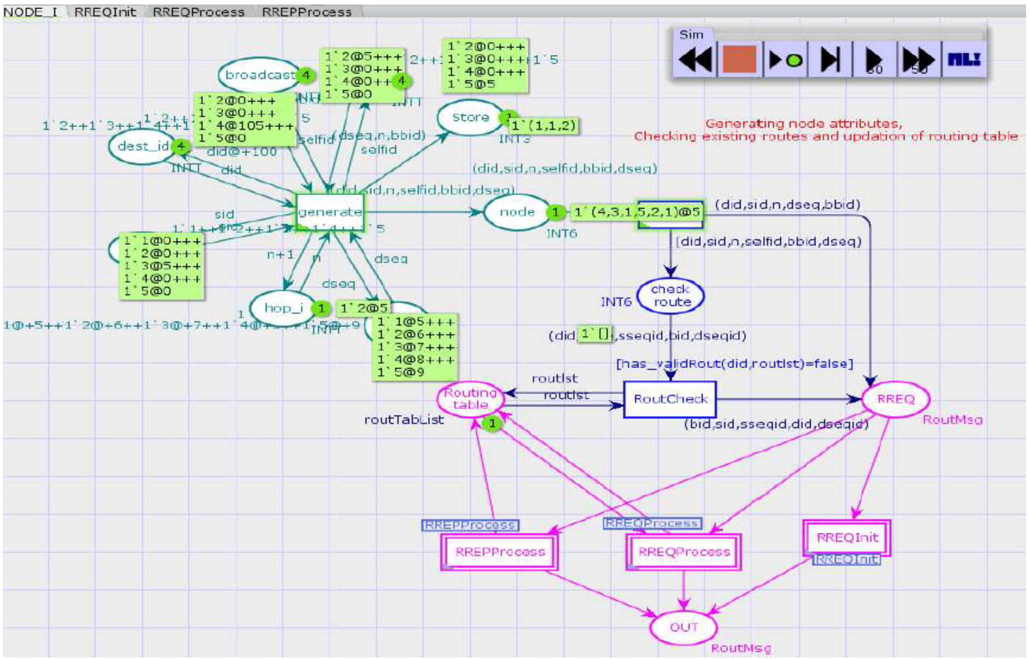


Figure 3.8: Simulation after the occurence of transition generate

Whenever transition ROUTECHECK is enabled, the guard function hasvalid route() is evaluated. A guard is a CPN ML Boolean expression that evaluates to TRUE or FALSE. Each expression used in a guard must be a Boolean expression. Here if the expression evaluates to false then transition is enabled and the packet moves to the next subpage i.e. RREQInit subpage.

In this subpage the transition BROADCAST is enabled and the tokens are fired with a time delay of 10units. A transition DELAY must be a positive integer expression. The expression is preceded by @+, and this means that the time inscription has the form @+ delay-expr.Time delay is always added relative to the current time. For example here current time is 5time units and the time delay is @+10 then the time stamp of tokens sent to the output places will be 15time units.

The token after moving from this subpage goes to the main page after which it is fired to RREQProcess subpage.

The tokens from the place RREQ moves to the place BIDCheck. Two paths exists from this transition. If the destination node mentioned is same as that of

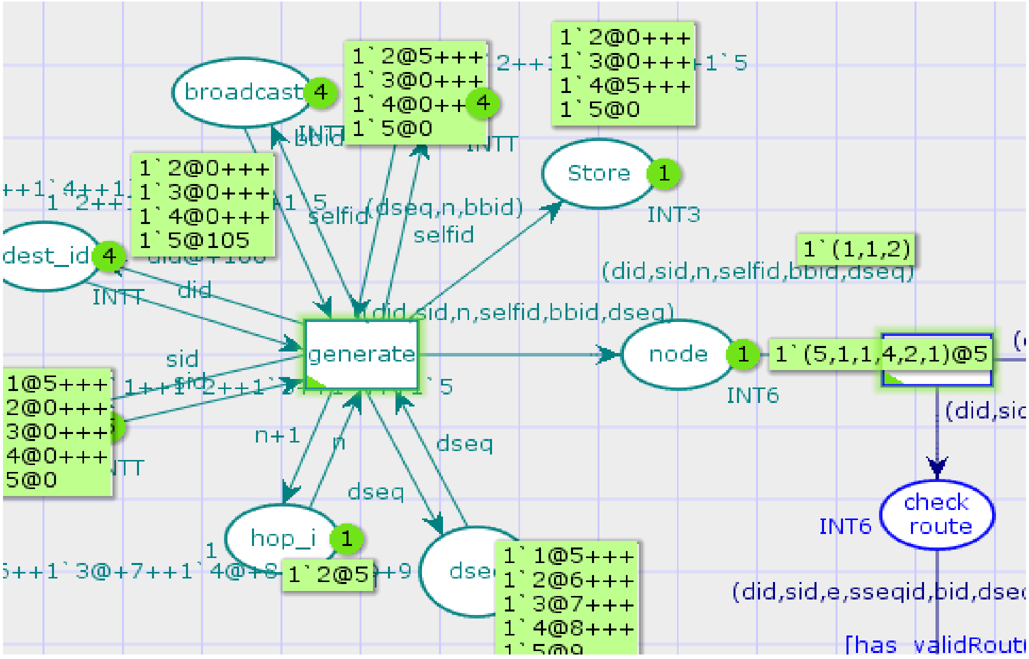


Figure 3.9: Simulation after the occurence of transition generate

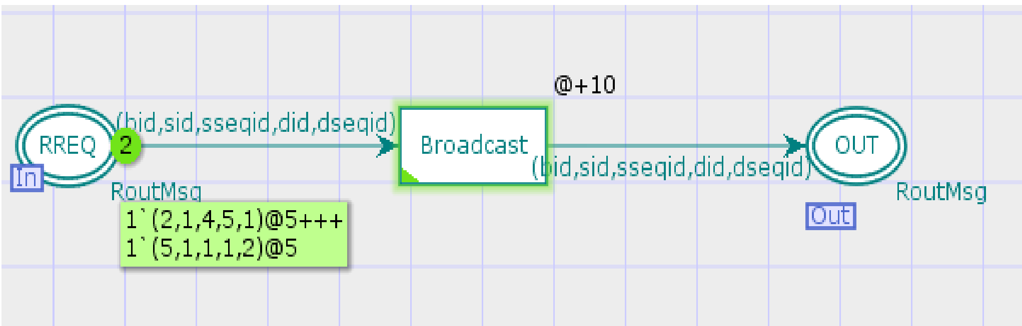


Figure 3.10: Simulation after the occurence of transition RoutCheck

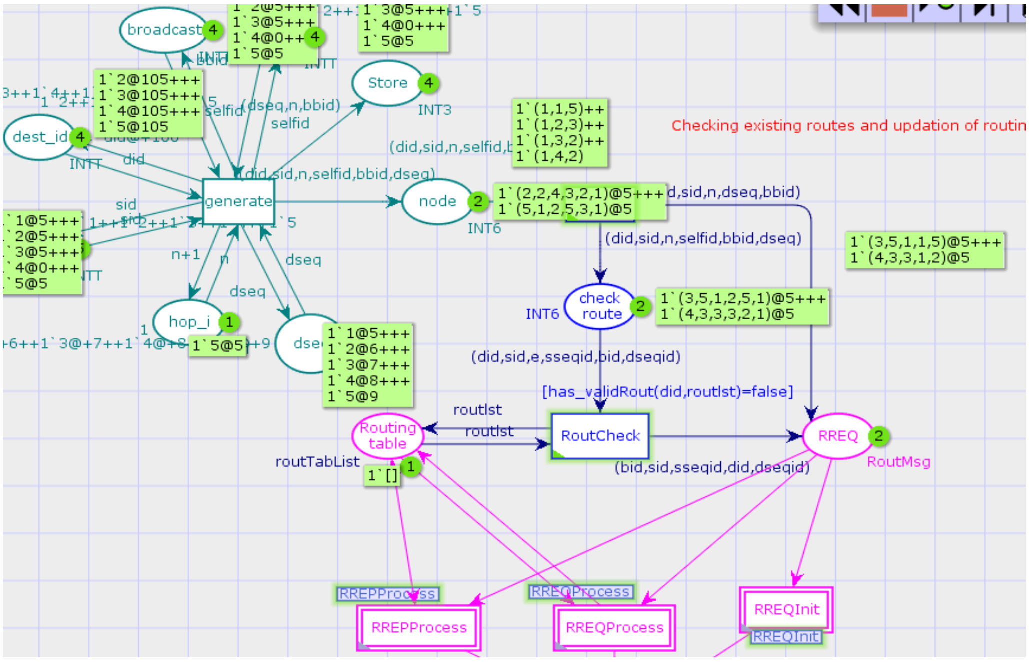


Figure 3.11: Simulation after the occurence of transition Broadcast

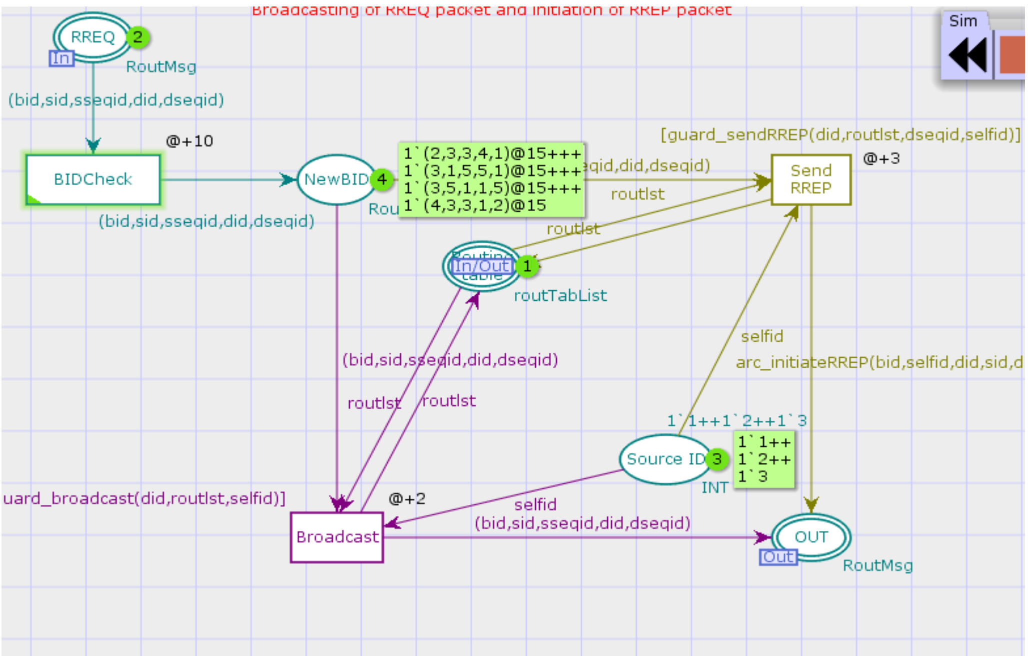


Figure 3.12: Simulation after the occurence of transition RoutCheck

self id then the packet moves towards the transition RREP otherwise same RREQ packet is broadcasted using the transition Broadcast. The transition SendRREP has a guard function guardsendRREP().As shown in Figure-3.13 this transition is enabled when did and self id are same like in this case its 3. Hence the transition is fired with addition of time delay of 3units as shown in Figure-3.13.

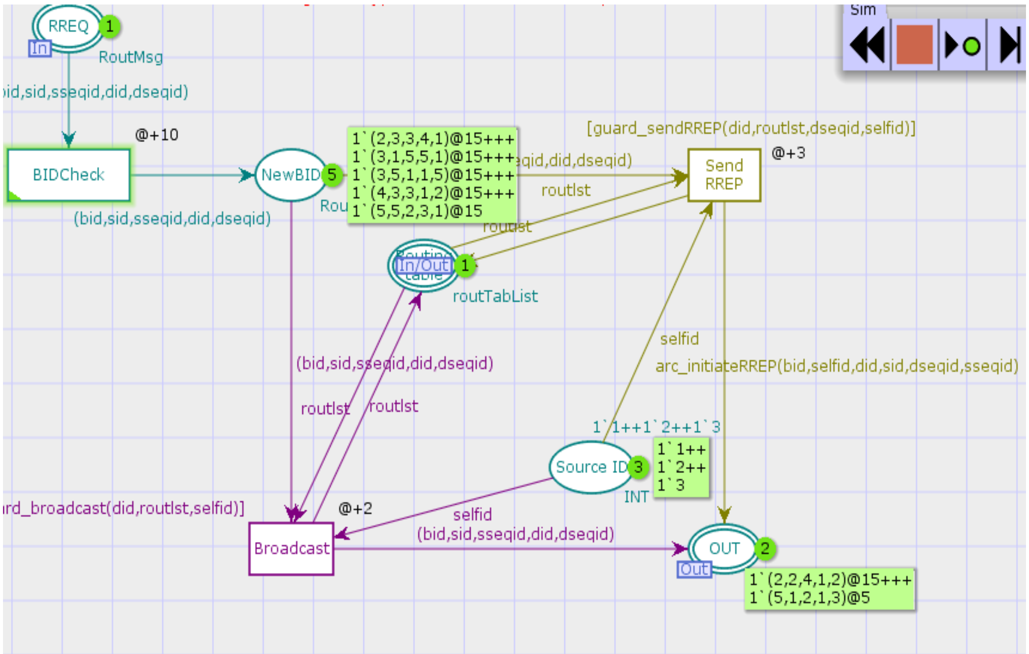


Figure 3.13: Simulation after the occurence of transition BIDCheck

Similarly the transition Broadcast has a guard guardbroadcast() which is enabled when the destination id and self id are not same. For example here the destination id is 2 and self id is 1, so the transition fires this token by adding time delay of 2units. The packet then moves to RREPProcess subpage where forwarding of RREP packet and calculation of delay takes place. The packet first moves to the transition AddRout(). As shown in Figure-3.16, the tokens from transition AddRout is transferred to two places OUT and receive token. When the token is transferred to the place OUT, the Routing table is updated using the arc inscription arcupdateRout(). When the tokens are available at place receivetoken(), the transition calculate delay is enabled. This transition is associated with a coding part where

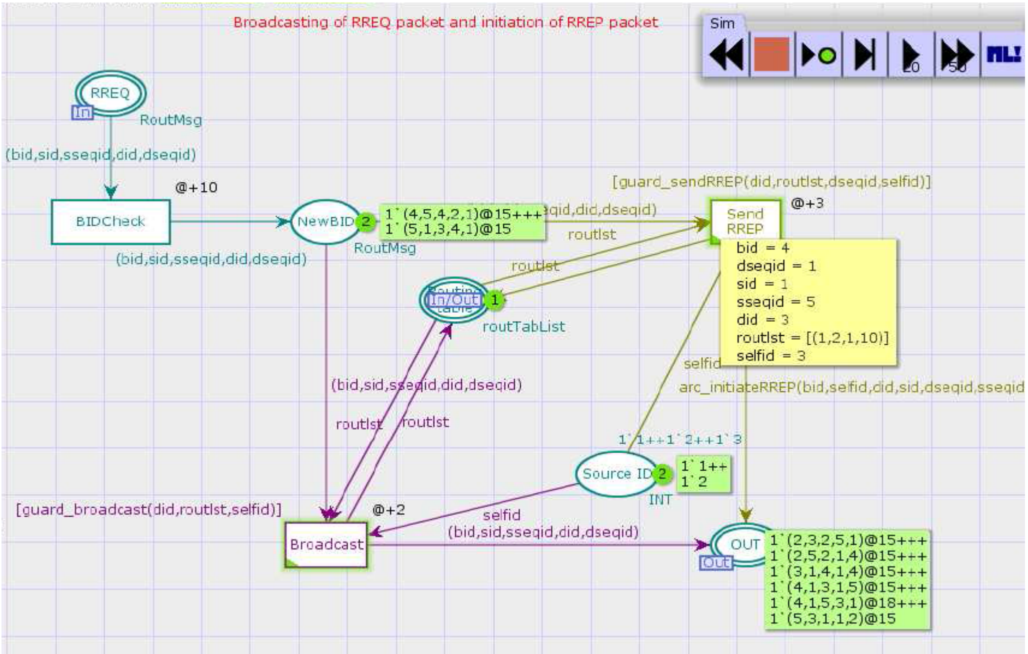


Figure 3.14: Tokens at transition SendRREP

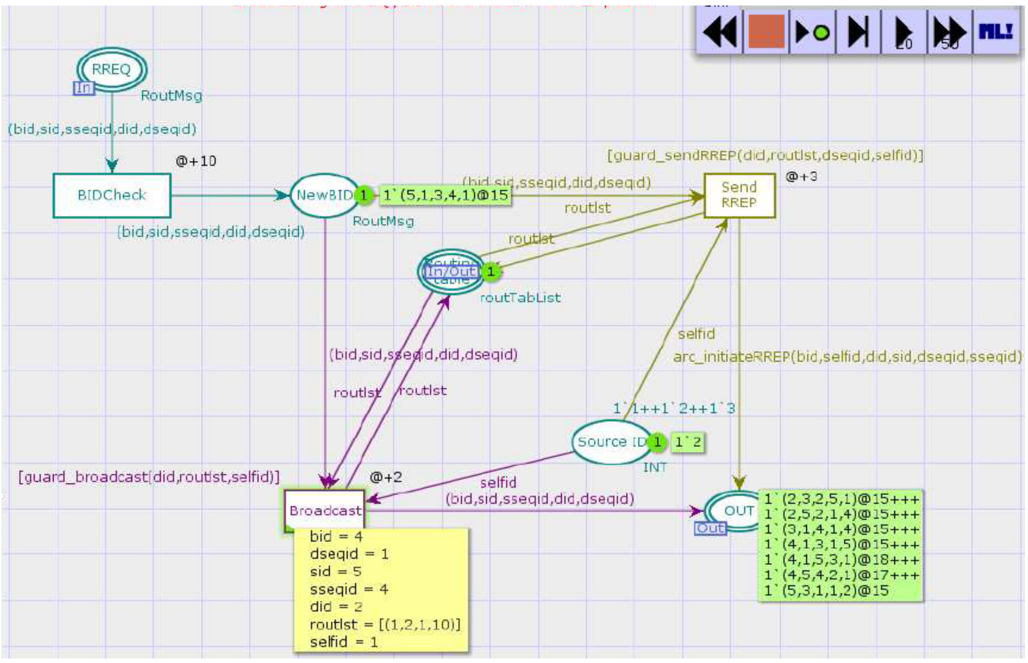


Figure 3.15: Tokens at transition Broadcast

the input part accepts the token and output part obtains the delay value associated with that token.

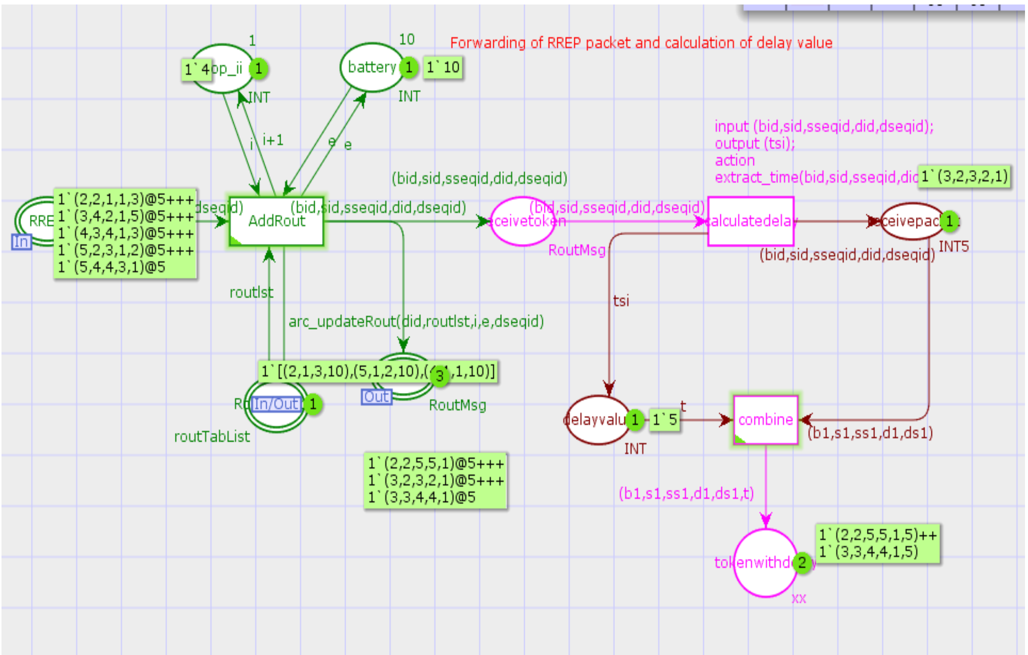


Figure 3.16: Simulation after the occurence of transition RoutCheck

Similarly rest of the tokens are also fired and all tokens moves to the place token with delay. This place contains all the tokens along with their delay values.

3.5 State Space analysis of AODV routing protocol using cpn tool

The SS instrument is coordinated with CPN Tools. This implies that you can undoubtedly switch between the supervisor, the test system, and the SS instrument. At the point when a state space hub has been discovered, it can be assessed in the test system. This implies that you can see the stamping specifically on the graphical representation of the CPN model. You can see the empowered move occasions, explore their ties and make reenactments. Similarly, when a checking has been found in the test system, it can be added to the state space or utilized as the introductory stamping for another state space.

The SS instrument has a substantial number of inherent standard questions. They can be utilized to explore all the standard properties of a CP-net, for example, reachability, boundedness, home properties, liveness and reasonableness. Notwithstanding the standard inquiries there are various capable hunt offices permitting you to define your own, non-standard questions. The standard questions oblige no programming by any stretch of the imagination. The non-standard questions typically oblige that you compose 2-5 lines of truly clear ML code.

CPN tool having a feature of state space palette. State space palette having different fields and each one was useful for creating a state space report and state space diagram. For applying state space palette we have to follow some rules at the time of modelling. Those are:

- ALL notations have names and names should be unique. Names of the notations should be follow the naming conventions.
- Follow the state space palette rules for applying state space.

The era of new hubs advances in an expansiveness first design. This implies that the hubs are prepared in the request in which they were made. To a certain degree, a profundity original can be acquired by utilizing "limited" Branching Options.

When you make an alteration of the CPN chart, it is important to recover all the state space code starting with no outside help. This additionally implies that the state space (if any) is lost. At the point when the alteration is made in the test system it is adequate to apply the Enter State Space apparatus once more.

The state space is ascertained for those parts of the net which would take an interest in a reproduction. If you don't mind take note of that, in state spaces, it just bodes well to utilize code fragments in an exceptionally restricted design, e.g., to initialise a CPN model.

Free variables on yield circular segments are not permitted ? unless they are variables of a little shading set.

When you have produced the state space code (by taking after the strides portrayed above), you are prepared to figure the state space.

In the event that the state space is required to be little (e.g., with a couple of hundred hubs and bends), you can essentially apply the Calculate State Space device, i.e. select the apparatus from the state space device palette and apply it to one of the pages of the net. Else you may need to change the Stop Options and/or the Branching Options portrayed.

A considerable lot of the question capacities utilize the Scc diagram (i.e., the emphatically joined parts of the state space). To ascertain the Scc chart you must apply the Calculate Scc Graph device, i.e. select the device from the state space apparatus palette and apply it to one of the pages in the net.

The state space and the Scc chart can likewise be computed by utilizing the Evaluate ML instrument to assess the accompanying ML capacities, which work precisely as the instruments in the state space device palette. This can, e.g., be valuable in the event that you need to handle exemptions raised by the CPN model or the Stop choice.

Our modelling should be satisfy the above rules before applying state space tool or palette for generating report and state space diagram. State space palette was showed in Figure-3.17. It having enter state space, calculate state space, SCC calculate. SS report, state space diagram and state indicators. By using the state space palette we can generate state space report for our modelling. State space report having statistics, Roundedness, home, liveness and fairness properties. State space report of AODV modelling was:

This state space report gives information about our modelling like loops, dead states, unreachable states and possible inputs of such states. In state space palette, we have a feature of display node with the specified number. It is useful for creating state space diagram shown in Figure-3.1824. In that page it gives information about

```
CPN Tools state space report for:
/cygdrive/E/thesis/mainproject.cpn
Report generated: Tue Oct 21 18:04:13 2014

 Statistics
------------------------------------------------------------------------

  State Space
     Nodes:  97602
     Arcs:   97601
     Secs:   304
     Status: Partial

  Scc Graph
     Nodes:  97602
     Arcs:   97601
     Secs:   28

 Boundedness Properties
------------------------------------------------------------------------

  Best Integer Bounds
                             Upper      Lower
     RREPProcess'battery 1   1          1
     RREPProcess'hop 1       1          1
     RREqprocess'new_BID 1   0          0
     RREqprocess'sourceid 1  3          3
     mainproject'broadcast 1 6          4
     mainproject'check_route 1
                             1          0
     mainproject'dest 1      4          4
     mainproject'dseq 1      5          5
```

Figure 3.17: state space report of AODV routing protocol

state also [1].

From the state space diagram and state space report we can easily identify dead states, loops and their inputs. We can also detect the reasons for occurring such kind of situations in network.

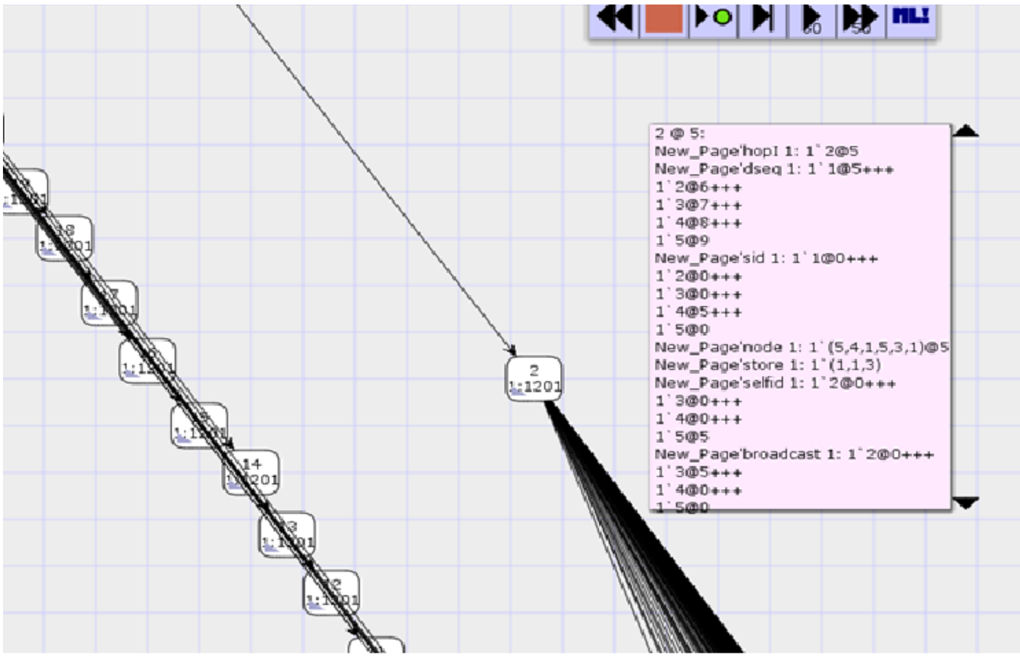


Figure 3.18: State Space diagram of AODV routing protocol

Chapter 4

Simulation in Colored Petri Net with help of NS2 and MATLab

In previous section, the AODV routing protocol is validated by providing the inputs manually. To improve the efficiency of simulation, the inputs are fed from a well known Network Simulation tool, NS2. The routing protocol is simulated by providing the scenario file as input. The delay is calculated using awk programming by obtaining the start time and end time from trace file. These values are modeled in CPN tool. The aimulator output remains same when the values are feeded through NS2. Figure-4.1 shows the overall process of work.

4.1 Simulation in NS2 tool

The aodv.tcl file is inputted to the ns-2 simulation tool. The file contains the code for existing AODV routing protocol. It is simulated in ns2 and it gives a trace file as its output. This trace file information described in Table-4.2 and Figure-4.2:

TABLE:Trace file information

S.no	What happened ????	Values for instance....
1	It shows the occured event	s- SEND, r- RECEIVED,D- DROPPED.
2	Time at which the event occurred	10.000000000.
3	Node at which the event occurred	Node id like 0.
4	Layer at which event occured	AGT- Application layer, RTR- Routing layer, LL -Link layer,IFQ- Interface Queue, MAC- Mac layer,ARP link layer.
5	Show flags	
6	Shows the sequence no of packets	0.
7	Shows the packet type	cbr- CBR packet, DSR- DSR packet, RTS- RTS packet generated by MAC layer.
8	Shows the size of packet	Packet size increases when a packet moves from an upper layer to lower layer and decreases when it moves from lower layer to an upper layer.
9	[....]	It shows the information about packet duration, mac address of destination,mac address of source and the mac type of the packet body.
10	Show flags	AGT- Application layer, RTR- Routing layer, LL -Link layer,IFQ- Interface Queue, MAC- Mac layer,ARP link layer.
- 11	[....]	It shows information about source node ip: port number, destination node ip(-1 means broadcast):port number,ip header ttl and ip of next hop(0 means node 0 or broadcast).

Table 4.1: Trace file information

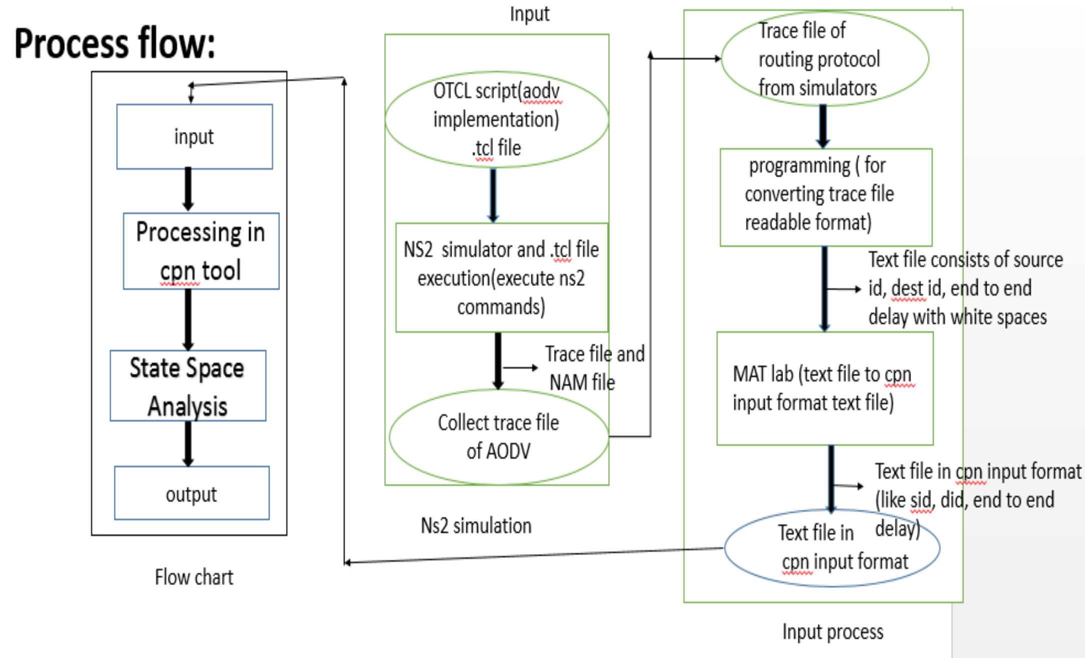
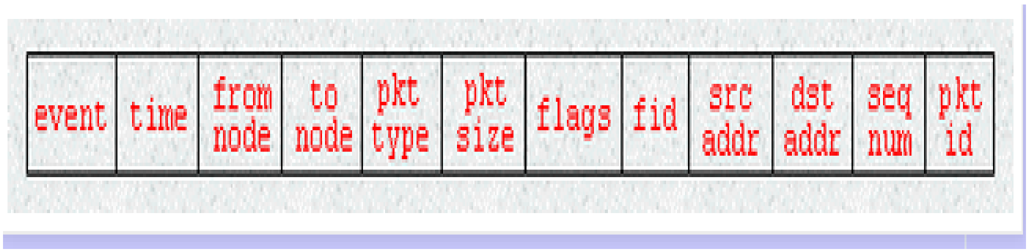


Figure 4.1: Flow chart depicting the steps involved

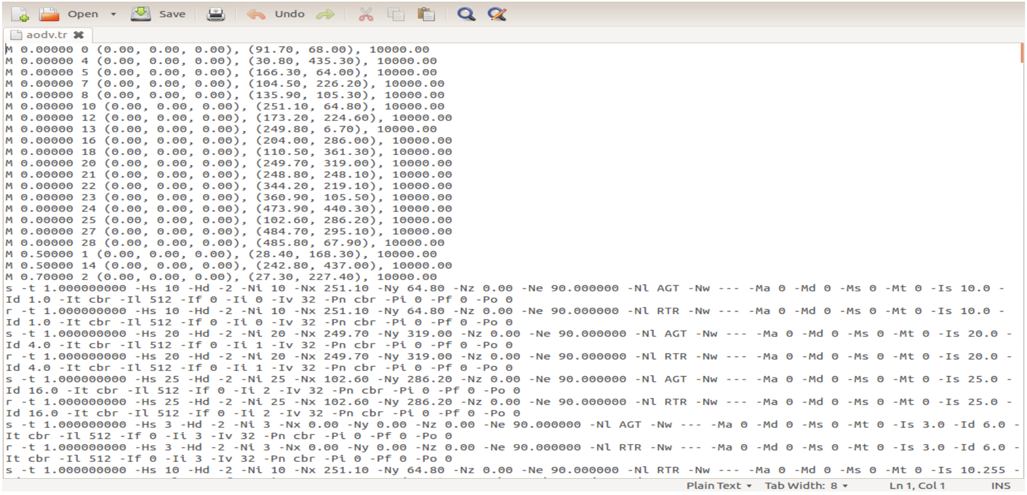


event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

Figure 4.2: trace information details

4.2 awk Programming

The trace file generated by simulating the code in ns2 tool contains the information as shown in the table. On the basis of these values we can extract the start time and end time for a particular packet by using the sequence no of the packet. The information is outputted to a text file. This file contains the source node, the destination node and the end to end delay. AWK program shown in Figure-4.4.



```
M 0.00000 0 (0.00, 0.00, 0.00), (91.70, 68.00), 10000.00
M 0.00000 4 (0.00, 0.00, 0.00), (30.80, 435.30), 10000.00
M 0.00000 5 (0.00, 0.00, 0.00), (166.30, 64.00), 10000.00
M 0.00000 7 (0.00, 0.00, 0.00), (104.50, 226.20), 10000.00
M 0.00000 8 (0.00, 0.00, 0.00), (135.90, 105.30), 10000.00
M 0.00000 10 (0.00, 0.00, 0.00), (251.10, 64.80), 10000.00
M 0.00000 12 (0.00, 0.00, 0.00), (173.20, 224.60), 10000.00
M 0.00000 13 (0.00, 0.00, 0.00), (249.80, 6.70), 10000.00
M 0.00000 16 (0.00, 0.00, 0.00), (204.00, 286.00), 10000.00
M 0.00000 18 (0.00, 0.00, 0.00), (110.50, 361.30), 10000.00
M 0.00000 20 (0.00, 0.00, 0.00), (249.70, 319.00), 10000.00
M 0.00000 21 (0.00, 0.00, 0.00), (248.80, 248.10), 10000.00
M 0.00000 22 (0.00, 0.00, 0.00), (344.20, 219.10), 10000.00
M 0.00000 23 (0.00, 0.00, 0.00), (360.90, 105.50), 10000.00
M 0.00000 24 (0.00, 0.00, 0.00), (473.90, 440.30), 10000.00
M 0.00000 25 (0.00, 0.00, 0.00), (102.60, 286.20), 10000.00
M 0.00000 27 (0.00, 0.00, 0.00), (484.70, 295.10), 10000.00
M 0.00000 28 (0.00, 0.00, 0.00), (485.80, 67.90), 10000.00
M 0.50000 1 (0.00, 0.00, 0.00), (28.40, 168.30), 10000.00
M 0.50000 14 (0.00, 0.00, 0.00), (242.80, 437.00), 10000.00
M 0.70000 2 (0.00, 0.00, 0.00), (27.30, 227.40), 10000.00
s -t 1.000000000 -Hs 10 -Hd -2 -Ni 10 -Nx 251.10 -Ny 64.80 -Nz 0.00 -Ne 90.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 10.0 -
Id 1.0 -It cbr -Il 512 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
r -t 1.000000000 -Hs 10 -Hd -2 -Ni 10 -Nx 251.10 -Ny 64.80 -Nz 0.00 -Ne 90.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 10.0 -
Id 1.0 -It cbr -Il 512 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 1.000000000 -Hs 20 -Hd -2 -Ni 20 -Nx 249.70 -Ny 319.00 -Nz 0.00 -Ne 90.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 20.0 -
Id 4.0 -It cbr -Il 512 -If 0 -Ii 1 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
r -t 1.000000000 -Hs 20 -Hd -2 -Ni 20 -Nx 249.70 -Ny 319.00 -Nz 0.00 -Ne 90.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 20.0 -
Id 4.0 -It cbr -Il 512 -If 0 -Ii 1 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 1.000000000 -Hs 25 -Hd -2 -Ni 25 -Nx 102.60 -Ny 286.20 -Nz 0.00 -Ne 90.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 25.0 -
Id 16.0 -It cbr -Il 512 -If 0 -Ii 2 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
r -t 1.000000000 -Hs 25 -Hd -2 -Ni 25 -Nx 102.60 -Ny 286.20 -Nz 0.00 -Ne 90.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 25.0 -
Id 16.0 -It cbr -Il 512 -If 0 -Ii 2 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 1.000000000 -Hs 3 -Hd -2 -Ni 3 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne 90.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3.0 -Id 6.0 -
It cbr -Il 512 -If 0 -Ii 3 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
r -t 1.000000000 -Hs 3 -Hd -2 -Ni 3 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne 90.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3.0 -Id 6.0 -
It cbr -Il 512 -If 0 -Ii 3 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
s -t 1.000000000 -Hs 10 -Hd -2 -Ni 10 -Nx 251.10 -Ny 64.80 -Nz 0.00 -Ne 90.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 10.255 -
```

Figure 4.3: AODV trace file

4.3 MATLAB programming

The text file produced using above awk programming contains the values separated by space. In order to make it readable by cpn tools, a matlab programming is done. It converts comma separated values into one which can be inputted to CPN tools.

4.4 Simulation using Colored Petri Net

The Figure-4.5 consists of a transition named INPUT where the code is written for inputting the values from a text file named outnew.txt. All the values inputted are accumulated in the place tokens. The vlaues inputted contains the source node,the destination node and the endtoend delay.

4.5 Effciency Calculation

The effciency of the Modeled AODV is calculated by taking into account the number of packets sent and the number of packets successfully received. The below table shows the comparison between Modeled AODV and NS2 AODV. The reason for the

```
# ==============================# AWK Script for calculating:
#       => Send
#       => Received
#       => Dropped packets, with packets
#       => Delivery Ration and Average End-to-End Delay.
BEGIN {
     seqno = -1;
     count = 0;
          }
         {
          if($4 == "AGT" && $1 == "s" && seqno < $6)
                  {
                   seqno = $6;
                  }
           if($4 == "AGT" && $1 == "s")
                  {
                          start_time[$6] = $2;
                          sn[$6]=$3;
                  } else if(($7 == "tcp") && ($1 == "r"))
                   {
                          end_time[$6] = $2;
                          dn[$6]=$3;
                  } else if($1 == "D" && $7 == "tcp")
                   {
                          end_time[$6] = -1;
                  }
}
END {
for(i=0; i<=seqno; i++)
{
          if(end_time[i] > 0)
          {
          delay[i] = end_time[i] - start_time[i];
          count++;
          print delay[i];
          print sn[i];
          print dn[i];
          print "\n";
           }else{
                   delay[i] = -1;
          }
}
```

Figure 4.4: AWK program for extracting the information from trace file

less effciency of the modeled CPN is that we are halting the simulation before all the packets have reached the destination. Therefore most of the packets are present in the network when we have taken the simulation output. We are halting simulation after a speci

ed number of steps because, the modeled AODV goes on generating packets in

nitively.

TABLE:NS2 AODV Effciency VS CPN AODV Effciency

Simulation .no	details	NS2 OUTPUT	AODV OUTPUT
Simulation 1	packet sent	56	56
	packet Received	38	36
	Throughput in percentage	67	65
Simulation 2	packet sent	170	170
	packet Received	150	145
	Throughput in percentage	88	85
Simulation 3	packet sent	814	814
	packet Received	774	740
	Throughput in percentage	95	91

Table 4.2: Efficient AODV Throughput comparision between NS2 and CPN

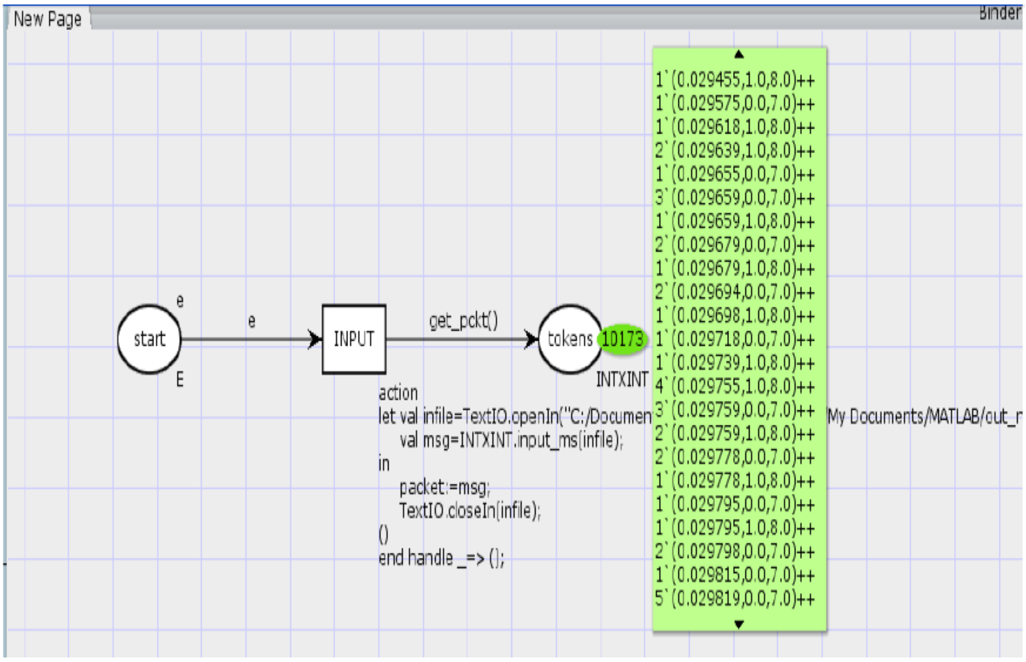


Figure 4.5: Inputting values into CPN tool

Chapter 5

Conclusion

In this thesis, AODV routing protocol is modeled using Colored petri net. From this thesis, we can conclude that CP nets are useful for validating the design, analysing the properties of model and functionalities. State space diagrams shows the loops, dead states and Boundedness of the nodes. CP nets provided an interface between the designers, developers and testers. It provides a view on implementation difficulties to designers. By using CP nets we can test the models from designing phase only. CP nets helps to detect the problems before occurring in real time and we can check our solutions weather it will work or not. We can surely say that, these models help to improve the success rate of development.

Scope for Further Research

Our work helps to detect the flaws in network protocols and flow of information in communication networks or routing protocols. CP nets are useful for verifying the design, validate the model weather is it possible to implement in real world or not. CP nets support for implementing a better design, product and provides the guidance for walkthroughs. Cp nets provides an interface between designers, developers (Coders) and testers.

Bibliography

[1] Deshraj Ahirwar, Abhishek Kesharwani, Sanjay Kumar Tehariya, and Azher Ahmed Khan. A secure routing approach of optimized link state routing protocol.

[2] J. Tsai C. Xiong, T. Murata. Modeling and simulation of routing protocol for mobile ad hoc networks using colored petri nets. *Foundations and Trends® in Networking*, 1(3):145–153, 2007.

[3] Gerhard Kramer, Ivana Marić, and Roy D Yates. Cooperative communications. *Foundations and Trends® in Networking*, 1(3):271–425, 2006.

[4] Carl Adam Petri and Wolfgang Reisig. Petri net. *Scholarpedia*, 3(4):6477, 2008.

[5] Lars M Kristensen, Soren Christensen, and Kurt Jensen. The practitioner?s guide to coloured petri nets. *International Journal on Software Tools for Technology Transfer (STTT)*, 2(2):98–132, 1998.

[6] Supriya Agrahari and Suchismita Chinara. Simulation of random waypoint mobility model using coloured petri nets. 2012.

[7] Fazli Erbas, Kyandoghere Kyamakya, and Klaus Jobmann. Modelling and performance analysis of a novel position-based reliable unicast and multicast routing method using coloured petri nets. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 5, pages 3099–3104. IEEE, 2003.

[8] L.M. Kristensen K. Jensen and L.Wells. The practitioner?s guide to coloured petri nets. *International Journal on Coloured Petri Nets and CPN Tools for modeling and validation of concurrent systems*, 2(2):98–132, 2007.

[9] Ruwini Kodikara, Sea Ling, and Arkady Zaslavsky. Evaluating cross-layer context exchange in mobile ad-hoc networks with colored petri nets. In *Pervasive Services, IEEE International Conference on*, pages 173–176. IEEE, 2007.

[10] Monika Heiner, Mostafa Herajy, Fei Liu, Christian Rohr, and Martin Schwarick. Snoopy–a unifying petri net tool. In *Application and Theory of Petri Nets*, pages 398–407. Springer, 2012.

[11] Kurt Jensen. A brief introduction to coloured petri nets. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 203–208. Springer, 1997.

[12] Lisa Wells. Performance analysis using cpn tools. In *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, page 59. ACM, 2006.

[13] Kurt Jensen and Lars M Kristensen. *Coloured Petri nets: modelling and validation of concurrent systems*. Springer Science & Business Media, 2009.

[14] Javad Mirabedini Ali Nakhaee, Ali Harounabadi. A novel communication model to improve mobile ad hoc network routing reliability using coloured petri nets. 2011.

[15] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[16] prof dr K M vanhee Wil M.P. van der Aalst, prof.dr. j. wessels. Timed coloured petrinets and their application to logistics. *Future Generation Computer Systems*, 26(8):1141–1150, 1992.

[17] Javad Mirabedini Ali Nakhaee, Ali Harounabadi. A novel communication model to improve mobile ad hoc network routing reliability. 2011.

[18] Elizabeth M Belding-Royer and Charles E Perkins. Evolution and future directions of the ad hoc on-demand distance-vector routing protocol. *Ad Hoc Networks*, 1(1):125–150, 2003.

[19] Himabindu Pucha, Saumitra M Das, and Y Charlie Hu. Imposed route reuse in ad hoc network routing protocols using structured peer-to-peer overlay routing. *IEEE Transactions on Parallel and Distributed Systems*, 17(12):1452–1467, 2006.

[20] C Siva Ram Murthy and BS Manoj. *Ad hoc wireless networks: Architectures and protocols*. Pearson education, 2004.

[21] Rendong Bai and Mukesh Singhal. Salvaging route reply for on-demand routing protocols in mobile ad-hoc networks. In *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–62. ACM, 2005.

[22] Harris Simaremare, A Syarif, A Abouaissa, RF Sari, and P Lorenz. Method for decreasing control messages in ad hoc networks. pages 47–51. Springer, 2006.

[23] Hongbo Zhou. A survey on routing protocols in manets. *Department of Computer Science and Engineering, Michigan State University, East Lansing, MI*, pages 48824–1027, 2003.

[24] Harris Simaremare, A Syarif, A Abouaissa, RF Sari, and P Lorenz. Energy consumption analysis of modified aodv routing protocol under random waypoint and reference point group mobility models. In *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*, pages 47–51. IEEE, 2012.

[25] Jian Liu and Fang-min Li. An improvement of aodv protocol based on reliable delivery in mobile ad hoc networks. In *Information Assurance and Security, 2009. IAS'09. Fifth International Conference on*, volume 1, pages 507–510. IEEE, 2009.